High Performance Lithography Hotspot Detection with Hierarchically Refined Machine Learning Methods

Duo Ding, *Andres J. Torres, *Fedor G. Pikus and David Z. Pan
ECE Dept. Univ. of Texas at Austin, Austin, TX 78712
*Mentor Graphics Corporation, 8005 S.W. Boeckman Road, Wilsonville, OR 97070
ding@cerc.utexas.edu, andres@mentor.org, fedor.pikus@mentor.com, dpan@cerc.utexas.edu

ABSTRACT
Lithography hotspot detection faces three challenges: 1) real hotspots are now harder to fix; 2) false alarm rate must be minimized; 3), full chip physical verification and optimization require fast turnaround. We propose a lithographic hotspot detection flow with ultra-fast speed and high fidelity that is especially suitable for guiding lithography-friendly design under real manufacturing conditions.

1. INTRODUCTION
With continuous shrinking of semiconductor process technology nodes, the minimum feature size of modern ICs is much smaller than the lithographic wavelength [1]. To bridge the wide gap between design demands and manufacturing limitations of the current mainstream 193nm lithography, various DFM techniques [2–4] have been proposed to improve product yield and avoid potentially problematic patterns (i.e., process hotspots). However, for 45nm node and below, hotspot patterns still exist even after design rule checking (DRC) and various resolution enhancement techniques (RET) such as optical proximity correction (OPC), sub-resolution assist feature insertions/layout re-targeting.

Therefore, fast, high fidelity hotspot detection engines can play an essential role to enhance physical verification/DRC (Design Rule Checking), and to develop process-aware physical design tools. Conventional approaches that employ lithographic simulations [5,6] are accurate, but very costly to run; on the other hand, approaches that utilize pattern/graph matching techniques [7–9] are fast, but they rely on a set of pre-defined hotspot patterns, which are generally hard to define and enumerate under real and ever more challenging manufacturing conditions.

In recent years, emerging works have started incorporating modern data mining methods for fast and accurate hotspot detection. A neural network judgment-based detection flow was proposed [10]. Data mining algorithms have been developed for hotspot pattern clustering [11]. While these early attempts have shown promising potential for data mining-based hotspot detection, there are still limitations yet to overcome, such as high training noise and low hotspot detection fidelity.

Further improvements were later made [12], where a support vector machine (SVM)-based hotspot detection method is utilized through performing 2D distance transform and histogram extraction. Also [13], SVM is employed for hotspot detection through classification of layout density metrics. However, issues with such approaches lie in run time and detection coverage, since 2D transforms and density extractions can be expensive to perform, while detection windows for the layout images are hard to anchor for full chip level detections. In [14], critical hotspot signature is proposed and extracted through certain special edge-based metrics. Although such edge-based extractions operate much faster compared with [12,13], its chip-level prediction still faces similar issues, such as scanning window coverage, etc.

Moreover, very few studies exist that deal with the detection challenges under the real manufacturing conditions that put stricter requirements to the detection engine to detect a small number of real hotspots with a low false alarm rate. Under such situations, the previously proposed approaches suffer from severe performance degradation.

In view of the current stage-of-the-art and the considerations of real manufacturing conditions, we sum up the key challenges of developing a high performance hotspot detection engine as follows:

- Hotspots are fewer, but harder to fix at post-layout stages, rendering the hotspot detection tasks into a much harsher environment.
- Non-hotspot detection accuracy becomes vitally important when the total number of patterns in a certain layout is enormously large; excessive false alarms lead to excessive penalty for hotspot correction.
- Large area layouts with densely placed and routed metal layers require ultra-fast detection speed.
- Sliding window techniques can be highly penalized by considerable accuracy loss due to coverage limitation.

To better address these challenges, we propose for the first time a hierarchically-refined machine learning framework for high performance hotspot detection, featuring the following main contributions:

- We define novel hotspot feature-centric characterization and powerful machine learning kernels for high fidelity detection under real manufacturing conditions.
- We introduce a hierarchically-refined machine learning-based flow for ultra-low false-alarm hotspot detection.
- We propose special hotspot signature measurements for ultra-fast, full layout detection without sliding window or raster scanning techniques.
- We perform thorough qualification using real industry examples for a 45nm metal1 process under real manufacturing conditions.

The rest of the paper is organized as follows. In Section 2, we introduce our proposed hotspot signature characterization and measurement for high detection coverage and speed. In Section 3, we describe an integrative flow with hierarchically refined classification techniques for ultra-low false-alarm hotspot detection under current real manufacturing conditions. Simulation results are presented and analyzed in Section 4. Section 5 concludes the paper.

This work was presented as a short paper for tutorial discussions at IEEE International Workshop on DFMkY, June 14th, 2010. For full length official publication/citation, please refer to ‘Duo Ding, Andres J. Torres, Fedor G. Pikus and David Z. Pan: High Performance Lithographic Hotspot Detection using Hierarchically Refined Machine Learning, Asia and South Pacific Design Automation Conference, Japan, Jan, 2011’
2. FEATURE-CENTRIC LAYOUT CHARACTERIZATION

Hotspot feature metrics (or hotspot signatures/models) refer to a set of special measurements that contribute strongly to successful decision-making processes for hotspot detections. The procedure to extract feature metrics from a design layout is referred to as layout context characterization. Unlike special restricted design rules, layout measurements and characterizations do not decide whether a certain pattern is defective or not, but leave the decision-making process to recursively trained and validated kernels (engines) using machine learning techniques. In face of the aforementioned challenges under real manufacturing conditions, a properly defined set of feature metrics plays the key role for a successful classification flow. Unlike previous studies utilizing 2D transforms, density-based calculations, or sliding window-based techniques, we propose novel metrics and special data structures for layout characterization with significant run-time reduction and satisfactory accuracy.

### 2.1 Hotspot Signature Measurement

Before introducing our special hotspot signature metrics, we define different types of feature measurements to capture characterize the layout with respect to: (a) corner information (convex or concave), (b) distance to an externally facing polygon edge, and (c) distance to an internally facing polygon edge. To carry out the feature measurements, we introduce four types of feature-centric operators constructed under the shared object environment of [15], as shown in Table 1. Consequently, hotspot-related geometries of a design layout are represented and indexed in high resolution (per-fragment-based) with a combination of the defined operators, leading to a high fidelity quantization process, as we will describe in detail in the following subsection.

### 2.2 Fragmentation-based Context Characterization

To provide full characterization coverage over the entire layout, our feature definition and extractions are carried out on a per fragment basis. Given a properly fragmented layout and any fragment of interest $F$, we illustrate the concept of an effective radius $r$ in our proposed context characterization procedure. As shown in Fig. 1, $r$ centers at (each) fragment $F$. By definition, effective radius $r$ covers the neighboring fragments that need to be considered in the context characterization of $F$. In its empirical nature, $r$ heavily depends on the lithography processes, and it is generally easy to pick in the training and validation procedure given the manufacturing conditions.

![Figure 1: Fragmentation-based hotspot signature extraction](image)

**Table 1: Hotspot signature measurement operators**

<table>
<thead>
<tr>
<th>operators</th>
<th>operation description (features to measure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{corn}()$</td>
<td>corner information: CV(convex) / CC(concave)</td>
</tr>
<tr>
<td>$f_{ext}()$</td>
<td>external inter fragment distances</td>
</tr>
<tr>
<td>$f_{int}()$</td>
<td>internal inter fragment distances</td>
</tr>
<tr>
<td>$f_{misc}()$</td>
<td>miscellaneous information</td>
</tr>
</tbody>
</table>

**Table 2: short-hand notation**

<table>
<thead>
<tr>
<th>notation</th>
<th>descriptions of the short-hand notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_i$</td>
<td>current fragment of interest (detection anchor point)</td>
</tr>
<tr>
<td>$F_{Ex}$</td>
<td>the fragment(s) facing $F$ externally</td>
</tr>
<tr>
<td>$F_{In}$</td>
<td>the fragment(s) facing $F$ internally</td>
</tr>
</tbody>
</table>

With Fig. 1(b) and Table 2, we describe the context characterization process in detail: suppose the current fragment of interest is $F$ (colored red in the center of the effective circle). We use $F_{In}$ to represent $F$’s internally-facing fragment, and $F_{Ex}$ for the externally-facing fragment. Similarly, $F_{ExIn}$ is the internally-facing fragment of $F_{Ex}$ and $F_{InEx}$ is the externally-facing fragment of $F_{In}$. Also, for each fragment, we mark its adjacent neighbors with indices in clockwise ascending order; for example, the first adjoining neighbor of $F$ in clockwise order is denoted as $F_{+1}$, and $F$ can also be denoted as $F_{0}$, as shown in Fig. 1(b). These indices are used for indexing throughout the fragments lying within the effective region’s projection onto certain polygon.

With all the notations explained, we present the characterized feature vector of fragment $F$ in the following formulation as in Eqn.(1)-(2):

$$ V_F = \sum_{i \in \delta F} \{f_{corn}(\bar{F}_i) \oplus f_{ext}(\bar{F}_i) \oplus f_{int}(\bar{F}_i) \oplus f_{misc}(\bar{F}_i)\} $$

$$ \bar{F} = [F, F_{Ex}, F_{In}, F_{ExIn}, F_{InEx}, ...] $$

where $F$ is an integer ID number representing a certain fragment in the layout, and $\delta F$ is the effective region of $F$. Operators $\oplus$ and $\sum$ are matrix operations for the generation of a vector $V_F$. The length of vector $V_F$ is the total number of features $M$.

In this paper, the parameter vector $V_F$ formed by the context characterization is defined as the hotspot signature metric, and the context characterization for each $F$ is also referred as the feature extraction process. Such a process filters out detection noise and provides a compact vector-based data set for machine learning kernels(MLK) to be properly established. Meanwhile, inside [15] environment, Eqn.(1) can be bulk processed with a one-time calculation for all fragments throughout the entire layout by directly invoking the afore-constructed operators, resulting in up to hundreds of times of run-time reduction compared with previous studies. Further discussions will be presented in Section 4.

### 3. HIERARCHICAL MACHINE LEARNING

#### 3.1 Overview

Due to the small number of real hotspots and the highly noisy detection environment under real manufacturing conditions, we hereby propose a hierarchically-refined machine learning method to integrate our proposed feature extraction processes and multi-level MLKs (Fig. 2,3). In its nature, such an approach hybrids the strength of ANN (or SVM) kernels and hierarchical classification methods, thus providing significant detection performance enhancement in terms
of run-time, detection accuracy, and reduced false alarms when compared with previous approaches using straightforward machine learning techniques.

3.2 Global Training and Detection

Here we refer to the first level training in Fig. 2 as the global training stage, since Model_1 is trained with the whole training dataset (on the global scale); similarly in Fig. 3, prediction with only Model_1 is defined as global detection. As we will show later in Section 4, under our proposed feature metrics and machine learning kernel implementations, the global detection stage alone achieves very satisfactory hotspot detection accuracy, whereas the non-hotspot detection accuracy is not high enough. Since hotspot and non-hotspot patterns are highly unbalanced under current real manufacturing conditions, the total non-hotspot number is so huge that even a small fraction of false alarms leads to excessive post-processing workload. Consequently, further refinements are required.

3.3 Hierarchical Local Refinement

To further suppress false alarms while maintaining a satisfactory hotspot detection rate, we extend the global stage with extra sub-levels, which we refer to as the local hierarchical refinement. As illustrated in Fig. 2 and Fig. 3, we employ hierarchical refinements in both the calibration and the prediction stages. In the calibration stage, the refinement flow consists of several key steps: (1) training and validating multiple hotspot metric models using the entire training data set and the false alarm data sets from a part of the validation data, (2) stopping criteria to decide when to stop adding more hierarchical models, (3) optimizations of the multiple thresholds \( \beta \) associated with the machine learning models. In the prediction stage, all the models and thresholds are employed, and hotspots are detected as those patterns that are identified as ‘positive’ in all hierarchies of models. We describe related key steps in more detail as follows:

Training and validating multiple models: Using the Model_1 from the global stage, the predict procedure is carried out over the combination set of the training data and a subset of the validation data. A variable threshold is properly chosen such that a proper set of false alarm patterns are derived. Note that the patterns within such a set are non-hotspots that are classified as hotspots by Model_1, and that, under real manufacturing conditions this set is usually large. Subsequently, an extra model Model_2 is derived by invoking ANN (or SVM) training process over the false alarm set. Similarly, such a routine continues by generating another false alarm set with Model_1/2 under threshold 1/2, leading to a third-level model Model_3, so on and so forth.

Stopping criteria: To quantify the stopping criteria for the multi-level calibration process, we introduce a user defined performance metric \( \Psi_{\text{perf}} \) in Eqn.(3):

\[
\Psi_{\text{perf}} = \alpha \cdot \text{Hit} + \beta \cdot \text{Nhit}
\]

where \( \alpha \) and \( \beta \) are user-defined weights, \( \text{Hit} \) is the hotspot detection accuracy and \( \text{Nhit} \) is the non-hotspot detection accuracy. Therefore, \( \Psi_{\text{perf}} \) represents the weighted summation of hotspot and non-hotspot detection accuracies. At every level, \( \Psi_{\text{perf}} \) is evaluated over \( \text{subset1} \) of the validation data, and the multi-level routine stops when \( \Psi_{\text{perf}} \) saturates or starts getting worse.

Thresholds optimization: We employ a heuristic approach for the threshold optimizations to exhaust the solution space with grid-based simulations and select the threshold combinations giving the best \( \Psi_{\text{perf}} \). The main reason is two fold: first, calibration process is performed only once in an \textit{a priori} manner and therefore does not lead to run-time overhead for predictions over testing layouts; second, based on our experiments in Section 4, \( \Psi_{\text{perf}} \) saturates or starts to deteriorate after level three, thus the total solution space is fairly limited. More details of this step can be found in Section 4.

Prediction and testing: Testing is carried out at speed with the calibrated models over a new set of layouts using the flow in Fig. 3, consisting of feature extraction and hierarchical machine learning classifications. Our proposed refinement detection flow differentiates the pattern streams and guides them through multiple levels of classification models, resulting in satisfactory detection accuracy and ultra-low false-alarms.

4. SIMULATIONS AND EXPERIMENTS

The simulation process involves three major steps: (1) under real manufacturing conditions, ANN- and SVM-based classification models are trained on a 500 \textit{um}² layout with fully placed and routed metal tracks in 45nm technology (in the global stage), (2) the validation (multi-level refinement)
owing to our ultra-fast hotspot signature characterization and non-hotspot detection accuracy. We also notice that while SVM kernels outperform ANN kernels in both hotspot result in faster runtime than modified SVM kernels, our proposed hierarchical detection flow, modified ANN kernel implementations into an integrative flow. We implemented our algorithm with an industry-strength engine under real manufacturing conditions in 45nm process, and demonstrated compelling performance improvement over previous studies in detection accuracy, false alarm rate and runtime. Such high performance lithographic hotspot detection under real manufacturing conditions is especially suitable for guiding lithography-friendly physical design.

5. CONCLUSION

We presented a high performance lithography hotspot detection flow (run-time and accuracy-wise) capable of provid-